

MySQL基础

主讲: Mr.King

Q Q: 382771946

邮箱: 382771946@qq.com

博客: <http://www.phpfamily.org>

微博: 新浪微博@Mr_King_Hs

MySQL介绍

“

Let's Begin

”

数据库

- 1.什么是数据库？
- 数据库(Database)是按照数据结构来组织、存储和关系数据的仓库。
- 2.什么是数据库系统？
- 数据库系统(database System)是由数据库及其管理软件组成的系统。
- 3.什么是数据库管理系统？
- 数据库管理系统(Database Management System)是一种操纵和管理数据库的大型软件，简称DBMS。



数据库

- 1.常见数据库MySQL、DB2 , SQL Server、 PostgreSQL等。
- 2.我们使用的是MySQL数据库。MySQL是一个开放源码的小型关联式数据库管理系统，开发者为瑞典MySQL AB公司。MySQL被广泛地应用在Internet上的中小型网站中。由于其体积小、速度快、总体拥有成本低，尤其是开放源码这一特点，许多中小型网站为了降低网站总体拥有成本而选择了MySQL作为网站数据库。



数据库

- 3.与其他的大型数据库例如Oracle、DB2、SQL Server等相比，MySQL自有它的不足之处，但是这丝毫也没有减少它受欢迎的程度。对于一般的个人使用者和中小型企业来说，MySQL提供的功能已经绰绰有余，而且由于MySQL是开放源码软件，因此可以大大降低总体拥有成本。Linux作为操作系统，Apache和Nginx作为Web服务器，MySQL作为数据库，PHP/Perl/Python作为服务器端脚本解释器。由于这四个软件都是免费或开放源码软件（FLOSS），因此使用这种方式不用花一分钱（除开人工成本）就可以建立起一个稳定、免费的网站系统，被业界称为“LAMP”组合。



MYSQL

- 1.在MySQL数据库中可以创建多个数据库，如果把每个数据库堪称是一个仓库，则网站中的内容数据就存储在这个仓库中，而对数据库中数据的存取及维护等，都是通过数据库软件管理的。同一个数据库管理系统可以为不同的网站分别建立数据库，但为了使网站中的数据便于维护，备份以及移植等，最好一个网站创建一个数据库。
- 2.MySQL数据库管理系统是一种C/S体系结构的管理软件，所以必须同时使用数据库服务器和客户端两个程序才能使用MySQL。服务器程序用于监听客户端的请求，并根据这些请求访问数据库，以便向客户端提供它们所要求的数据。而客户端程序必须通过网络链接到数据库服务器，才能向服务器提交数据操作请求。



SQL结构化查询语言

- 1.对数据库服务器中的数据的管理，必须使用客户机程序成功连接以后再通过必要的操作指令对其进行操作，这种数据库操作指令被称为SQL(Structured Query Language)语言，即结构化查询语言。MySQL支持SQL作为自己的数据库语言，SQL是一种专门用于查询和修改数据库里的数据，以及对数据库进行管理和维护的标准化语言。SQL语言结构简洁，功能强大，简单易学，所以应用广泛。
- 2.SQL语言包含四个部分：
- DDL、DML、DQL、DCL



SQL结构化查询语言

- 1.DDL(数据定义语言[Data Defination Language])用于定义和管理数据对象，包括数据库、数据表等。如：CREATE,DROP,ALTER
- 2.DML(数据操纵语言[Data Manipulation Language])用于擦做数据库对象中所包含的数据。如：INSERT,UPDATE,DELETE语句。
- 3.DQL(数据查询语言[Data Query Language])用于查询数据库对象中所包含的数据，能够进行单表查询，连接查询，嵌套查询以及集合查询等各种复杂程度不同的数据库查询。如：SELECT
- 4.DCL(数据控制语言[Date Control Language])用来管理数据库的语言，包括管理权限及数据更改。如：GRANT，REVOKE，COMMIT，ROLLBACK等。



MySQL操作

- 1.登陆
- Mysql -h服务器主机地址 -u用户名 -p用户密码
- 2.退出
- \q、exit、quit、ctrl+c退出当前客户端
- 注意：
- 每个SQL命令都需要使用分号完成
- 可以将一行指令拆成多行命令
- 可以通过\c取消本行命令不执行
- 3.查看帮助
- \h或者?来查看帮助(要学会差帮助)



MySQL操作

- 1.开启输出文件
- \T 日志文件的位置及名称
- 2.关闭输出文件
- \t
- 3.打印MySQL版本信息
- `Select version();`
- 4.打印当前服务器的日期时间
- `Select now();`
- 5.打印当前打开的数据库
- `Select database();`



数据库使用

- 1.创建数据库语法格式：
- `CREATE {DATABASE|SCHEMA} [IF NOT EXISTS] db_name [DEFAULT] CHARACTER SET [=] charset_name ;`
- 2.显示所有数据库
- `SHOW DATABASES;`
- 3.查看创建好的数据库
- `SHOW CREATE DATABASE db_name;`
- 4.打开数据库
- `USE db_name;`
- 5.修改数据库
- `ALTER DATABASE db_name [DEFAULT] CHARACTER SET [=] charset_name;`
- 6.删除数据库
- `DROP DATABASE [IF EXISTS] db_name;`



数据库使用

- 注意：
- 1.在Windows下，数据库名称不区分大小写，在Linux下区分，所以我们就统一遵守大小写
- 2.创建一个数据库就会在data目录下创建一个以数据库名称命名的文件夹
- 3. MYSQL数据库中命令不区分大小写。



数据表

- 数据表是数据库中的基本对象元素，以记录(行)和字段(列)组成的二维结构用于存储数据。数据由表结构和表内容两部分组成，先简历表结构，然后才能输入数据。数据表结构设计主要包括字段名称、字段类型、字段属性。同一个数据库中可以存储多个数据表，但是每个数据表名唯一，表中的每一条记录描述了一个相关信息的集合，每一个字段必须为唯一的，每个字段都需要指定数据类型。



数据表的使用

- 1.创建数据表
- CREATE TABLE [IF NOT EXISTS] table_name(
 字段名称 数据类型[UNSIGNED|ZEROFILL] [NULL|NOT NULL] [DEFAULT默认值][AUTO_INCREMENT] [UNIQUE[KEY]] | [PRIMARY] KEY

•)ENGINE= '引擎' ,DEFAULT CHARSET [=] "编码" ;
- 注意：
- 1.NULL指字段值允许为空，即在插入值时没有在此字段插入值，默认值为NULL值
- NOT NULL指字段值不允许为空，也就是插入值时必须给此字段插值



数据表的使用

- 2.UNSIGNED只能用于设置数值类型，不允许出现负数，最大存储长度会增加一倍，也就是无符号的存储范围，ZEROFILL只能用于设置数值类型，在数值钱会自动用0补齐不足的位数。
- 3.DEFAULT 默认值指如果在插入记录时，没有明确为该字段赋值，则MySQL将自动把默认值赋值该字段。
- 4.不能为空且没有默认值的字段在插入记录时必须赋值
- 5.允许为空的字段，在插入记录时，字段值可以用NULL表示
- 6.存在默认值的字段，在插入记录时，字段值可以用default表示



数据表的使用

- 7.AUTO_INCREMENT指字段值可以自动编号，只用于数值类型的字段，自动编号的字段，在插入记录时既可以人为赋值(一定不能存重复值),也可以采用MySQL的自动编号值，还可以采用null或default来表示
- 8.一张数据表只能存在一个自动编号的字段，且必须定义为主键
- 9.UNIQUE [KEY],唯一约束
- 10.[PRIMARY] KEY，主键约束，每张数据表只能存在一个主键约束，主键约束可以保证记录的唯一性，而且自动的禁止为空。
- 11.UNIQUE [KEY]唯一约束，保证记录的唯一性，每张数据表可以存在多个唯一约束，该字段值可以为空。



字段类型

- 1.数值类数据列类型
- 2.字符串类数据列类型
- 3.日期和时间类数据列类型
- 4.NULL值



数值类数据列类型

数据列	存储空间	说明	取值范围
TINYINT	1字节	非常小的整数	带符号值: -128~127 无符号值: 0~255
SMALLINT	2字节	较小的整数	带符号值: -32768~32767 无符号值: 0~65535
MEDIUMINT	3字节	中等大小的整数	带符号值: -8388608~8388607 无符号值: 0~16777215
INT	4字节	标准整数	带符号值: -21亿~21亿 无符号值: 0~42亿
BIGINT	8字节	大整数	带符号值: -2的63~2的63-1 无符号值: 0~2的64-1



数值类数据列类型

数据列	存储空间	说明	取值范围
FLOAT	4或8字节	单精度浮点数	最小非零值: $\pm 1.175494351 \times 10^{-38}$ 最大非零值: $\pm 3.402823466 \times 10^{38}$
DOUBLE	8字节	双精度浮点数	最小非零值: $\pm 2.225073 \times 10^{-308}$ 最大非零值: $\pm 1.797693 \times 10^{308}$
MEDIUINT	3字节	中等大小的整数	带符号值: -8388608~8388607 无符号值: 0~16777215
DECIMAL	自定义	以字符串形式表示浮点数	取决于存储单元



数值数据列类型

- 注意：
- 1.INT(3)、SMALLINT(3)等整型后面的数字不会影响数值的存储范围，只会影响显示
- 2.整型后面括号中的数字只有配合零填充的时候才有实际意义
- 3.整型后面括号中的数字可以省略
- 4.浮点型后面的数字会将存入的数字四舍五入，如把一个1.234存入FLOAT(6,1)数据列中，结果是1.2，6代表显示长度，1代表小数位长度，会四舍五入。



字符串类数据列类型

数据列类型	存储空间	说明	取值范围
CHAR[(M)]	M字节	定长字符串	M字节
VARCHAR[(M)]	L+1字节	可变字符串	M字节
TINYBLOB TINYTEXT	L+1字节	非常小的BLOB(二进制大对象)和文本串	2的8-1字节
BLOB,TEXT	L+2字节	小BLOB和文本串	2的16-1字节
MEDIUMBLOB MEDIUMTEXT	L+3字节	中等的BLOB和文本串	2的24-1字节



字符串类数据列类型

数据列类型	存储空间	说明	取值范围
LOB LONGTEXT	L+4字节	大LOB和文本串	2的32-1
ENUM('val1','val2'...)	1或者2字节	枚举：可赋予某个枚举成员	65535个成员
SET('val1','val2'...)	1,2,3,4或8字节	集合：可赋予多个集合成员	64个成员



字符串数据列类型

注意：

- 1.CHAR和VARCHAR类型长度范围都在0~255之间
- 2.在使用CHAR和VARCHAR类型时，当我们传入的实际的值的长度大于指定的长度，字符串会被截取至指定长度
- 3.在使用CHAR类型时，如果我们传入的值的长度小于指定长度，实际长度会使用空格补至指定长度
- 4.在使用VARCHAR类型时，如果我们传入的值长度小于指定长度，实际长度即为传入字符串的长度，不会使用空格填补
- 5.CHAR要比VARCHAR效率更高，但占用空间较大



字符串数据列类型

注意：

6.BLOB和TEXT类型是可以存放任意大数据的数据类型

7.BLOB区分大小写，TEXT不区分大小写

8.ENUM和SET类型是特殊的串类型，其列值必须从固定的串集中选择

9.ENUM只能选择其中一个值，SET可以选择多个值



时间和日期类数据类型

数据列类型	存储空间	说明	取值范围
DATE	3字节	“YYYY-MM-DD”格式表示的日期值	1000-01-01~9999-12-31
TIME	3字节	“hh:mm:ss”格式表示的时间值	-838:59:59~838:59:59
DATETIME	8字节	“YYYY-MM-DD hh:mm:ss”格式	1000-01-01 00:00:00~9999-12-31
TIMESTAMP	4字节	“YYYYMMDDhhmmss”格式	1970-0-01~2037年的 某个时刻
YEAR	1字节	“YYYY”格式的年份值	1901~2155

注意：存储日期，我们用整型来进行存储时间戳，便于日期计算



NULL值

NULL值使用:

- 1.NULL意味着没有值或者未知值
- 2.可以测试某个值是否为NULL，不能通过值==NULL来测试，通过is NULL
- 3.不能对NULL值进行算数计算，即使NULL进行算数运算结果还是NULL
- 4.0或NULL都意味着假，其余值都意味着真



DDL操作

- 1.ALTER DATABASE用于修改数据库的全局特性。这些特性储存在数据库目录中的db.opt文件中
- ALTER [DATABASE|SCHEMA] db_name [DEFAULT] CHARACTER SET [=] '字符编码' ;
- 2.ALTER TABLE 用于修改原表的结构。可以增加或删减列(也就是字段)，创建或取消索引，更改原有列的类型，或重新命名列或表，还可以更改表的评论和表的类型。
- ALTER TABLE table_name ...
- 1》添加字段
- ALTER TABLE table_name ADD 字段名称 字段类型 [first|after]字段名



DDL操作

- 2》添加主键(Primary key)
- ALTER TABLE table_name ADD PRIMARY KEY(字段名称)
- 3》添加唯一
- ALTER TABLE table_name ADD UNIQUE(字段名称 , 字段名称...)
- 4》添加外键
- ALTER TABLE table_name ADD CONSTRAINT 外键名称
FOREIGN KEY(字段名称) REFERENCES 主表名(字段名称)
- 5》修改字段名称或者修改字段名和类型
- ALTER TABLE table_name CHANGE 原字段名 新字段名 字段类型



DDL操作

- 6》修改字段类型
- ALTER TABLE table_name MODIFY 字段名 字段类型
- 7》修改或者删除字段的默认值
- 修改默认值：ALTER TABLE table_name alter 字段名 set default '值'
- 删除默认值：ALTER TABLE table_name alter 字段名 DROP DEFAULT;
- 8》删除字段列
- ALTER TABLE table_name drop 字段名;
- 9》删除主键
- ALTER TABLE table_name DROP PRIMARY KEY;



DDL操作

- 10》删除外键
- ALTER TABLE table_name DROP FOREIGN KEY 约束名称;
- 11》添加索引
- ALTER TABLE table_name ADD INDEX 索引名称(字段名称);
- 12》删除索引
- ALTER TABLE table_name DROP INDEX 索引名称;
- 13》禁用索引
- ALTER TABLE table_name DISABLE KEYS;
- 14》启用索引
- ALTER TABLE table_name ENABLE KEYS;



DDL操作

- 15》更改表名
- ALTER TABLE table_name RENAME [TO] 新表名;
- 16》根据某个字段来排序
- ALTER TABLE table_name ORDEY BY 字段名 ASC(升序)|DESC(降序);
- 17》转换字符编码
- ALTER TABLE table_name CONVERT TO CHARACTER SET '字符编码' ;
- 18》创建数据库
- CREATE DATABASE|SCHEMA [IF NOT EXISTS]数据库名称[[DEFAULT] CHARACTER SET [=] '字符编码'] ;



DDL操作

- 19》创建数据表
- CREATE [TEMPORARY] TABLE [IF NOT EXISTS] table_name(
• 字段名 字段类型 ...
•);
- 20》删除数据库
- DROP DATABASE|SCHEMA [IF EXISTS] db_name ;
- 21》删除数据表
- DROP TABLE [IF EXISTS] table_name ;
- 22》创建索引
- CREATE [UNIQUE|FULLTEXT|SPATIAL] INDEX 索引名称 on table_name(字段名称)



DDL操作

- 23》删除索引
- DROP INDEX 索引名称 ON table_name;
- 24》重命名表名
- RENAME TABLE 原表名 TO 新表名;



DML操作

- 1.插入数据
- Insert [into] table_name [(字段名 , 字段名...)] values('值' ; 值' ...)
- 或者
- Insert [into] table_name[(字段名,字段名)]
values('值' , '值' ..),('值' ; 值'),...
- 2.更新数据
- Update table_name 字段名= '值' [,字段名= '值'][where 条件]
[order by 字段] [limit 行数]
- 如果不指定where条件将更新全表数据



DML操作

- 3.删除数据
- Delete from table_name [where 条件] [order by 字段] [limit 行数]
- 4.清空数据表
- Truncate table table_name;



DQL操作

- 语法：
- SELECT 列名1 [AS] 别名，列名2,...列名n FROM table_name WHERE 子句
GROUP BY 子句（分组统计） HAVING 子句 ORDER BY 子句 LIMIT 子句
- 注意：
- 1.查询时尽量避免使用*,而要针对需求查询指定字段
- 2.为了解决多表之间字段名冲突，可以通过as取别名，as也可以省略
- 3.可以使用distinct去掉重复记录，但是distinct是作用在整个表的数据，所以想使用distinct的时候最好select *,count(distinct 字段) from table_name group by 字段;
- 或者直接通过分组
- Select * from table_name group by 字段;



DQL操作

- 4.使用where条件查询时可以结合使用逻辑运算符和比较运算符结合使用。
- 逻辑操作符：

操作符	语法	描述
AND 或 &&	a AND b 或 a && b	逻辑与，若两个操作数同时为真，则为真
OR 或	a OR b 或 a b	逻辑或，只要有一个操作数为真，则为真
XOR	a XOR b	逻辑异或，若有且仅有一个操作数为真，则为真
NOT 或 !	NOT a 或 !a	逻辑非，若操作数为假，则为真



DQL操作

- 比较运算符：

数据列类型	存储空间说明	取值范围
=	a=b	若操作数a与操作数b相等，则为真
<=>	a<=>b	若a与b相等，则为真，可以用于NULL值比较
!=或<>	a!=b或a<>b	若操作数a与b不相等，为真
<	a<b	若操作数a小于b，为真
<=	a<=b	若操作数a大于b，为真
>	a>b	若操作数a大于b，为真
>=	a>=b	若操作数a大于等于b，为真
IS NULL	a IS NULL	若操作数a为NULL，为真
IS NOT NULL	a IS NOT NULL	若操作数a不为NULL，为真
BETWEEN	a BETWEEN b AND c	若a在b和c之间，为真



DQL操作

- 比较运算符：

数据列类型	存储空间说明	取值范围
NOT BETWEEN	a NOT BETWEEN b AND c	若操作数a不在b和c之间，为真
LIKE	a LIKE b	SQL模式匹配，若a匹配b，为真
NOT LIKE	a NOT LIKE b	SQL模式匹配，若a不匹配b，则为真
IN	a IN (b1,b2....)	若a等于b1,b2,b3,...中的某一个，则为真



DQL操作

- 5.NULL值只能用is null或者is not null来检索。
- 6.对范围搜索的时候可以使用比较运算符>=和<=，也可以使用between and来实现。
- 7.使用in关键字进行查询，在()中放入查询值，至少一个值，多个值用逗号分隔
- 8.使用like进行模糊查询，查询条件中的百分号%表示0个或任意多个字符；下划线_表示单个的任意字符。不匹配使用not like。



DQL操作

- 连接查询
- 1》内连接(inner join):只连接匹配的行
- Table_name [INNER] JOIN 数据表 ON 连接条件
- 2》外连接
- 左外连接(left outer join):包含左边表的全部行（不管右边的表中是否存在与他们匹配的行），以及右边表中全部匹配的行
- 右外连接（right outer join）：包含右边表的全部行（不管左边表中是否存在与他们匹配的行），以及左边表中全部匹配的行
- Table_name {left[outer]join|right [outer] join} 数据表 on 连接条件



DQL操作

- 3》全外连接(full outer join):包含左右两个表的全部行，不管另外一边的表中是否存在与他们匹配的行。
- 4》自身连接：是指同一个表自己与自己进行连接（无限极分类）



MYSQL中查用函数

- 1.数值函数
- Rand():取0~1之间随机数
- 2.日期函数
- 当前日期 : curdate(),current_date,current_date();
- 当前时间 : curtime(),current_time,current_time();
- 当前日期时间 : now(),sysdate();
- 3.字符串函数
- 字符串长度 : LENGTH
- SELECT LENGTH("THIS IS A TEST");



MYSQL中查用函数

- 连接字符串CONCAT
- `SELECT CONCAT("KING" , " HS");`
- 聚合函数：
- Min:最小值
- Max：最大值
- Avg：平均值
- Sum：求和
- Count():记录个数

